EC-LoRA: Energy-Driven Continual LoRA Implicit Generation

Anonymous authors
Paper under double-blind review

Abstract

Low-Rank Adaptation (LoRA) enables lightweight adaptation of large language models (LLMs), but continual adaptation across tasks still requires maintaining and selecting task-specific adapters. To address this issue, we propose **EC-LoRA**, an energy-guided implicit generative framework for producing LoRA modules in a continual learning setting. Instead of directly decoding adapter weights with an explicit generator, EC-LoRA learns a task-conditioned energy function over the normalized parameter space. For each task, we initiate a prototype and generate a task-specific adapter by refining via energy-based gradient descent. The energy model is trained with score matching and updated sequentially across tasks. Our method generates only LoRA and reuses the non-LoRA classification or regression head from a selected source checkpoint, making it applicable to tasks with heterogeneous output spaces. We conducted experiments to verify that the LoRA generated by EC-LoRA outperforms the original LoRA and can adapt to different models and scenarios.

1 Introduction

Large Language Models (LLMs) have become increasingly capable general foundational model, but adapting to downstream tasks remains computationally and organizationally expensive (Devlin et al., 2019; Brown et al., 2020; Touvron et al., 2023). Parameter-efficient fine-tuning methods, such as LoRA (Hu et al., 2022), reduce this cost by updating only a small set of adapter parameters while keeping the base model frozen (Houlsby et al., 2019; Li & Liang, 2021; Lester et al., 2021; Zaken et al., 2022). In practice, however, multi-task and continual adaptation still requires selecting and evaluating many LoRA. As the number of tasks grows, the adapter space itself becomes a target of modeling: rather than training each LoRA checkpoint as an isolated artifact, we ask whether the distribution of high-performing LoRA parameters can be learned or optimize and used to generate task-specific adapters instead of retraining.

A direct approach is to train an explicit generator or hypernetwork that maps a task representation to LoRA weights (Ha et al., 2016; Mahabadi et al., 2021; Lv et al., 2024; Shao et al., 2025b). While appealing, such a formulation requires the model to decode a high-dimensional LoRA vector in one forward pass. Moreover, in continual learning setting, the generator must incorporate new tasks without overwriting the parameter regions associated with earlier tasks (Kirkpatrick et al., 2017; Zenke et al., 2017; Rusu et al., 2016; Chaudhry et al., 2019). These challenges suggest that LoRA generation should not only predict a point estimate, but also model the geometry of high-quality adapter parameters.

To address the issue, we propose **EC-LoRA**, an energy-driven implicit generative framework. Instead of directly decoding LoRA weights, EC-LoRA builds up a task-conditioned energy-based models over the low dimension of LoRA distribution. For each observed task, we construct a task prototype by available conditional LoRA checkpoints. At inference time, generation starts from the prototype and performs gradient-based refinement under the learned energy-based model. The high-quality task-specific LoRA is therefore produced by an implicit generation process.

For EC-LoRA, tasks arrive sequentially, and a single shared energy model is updated across the task sequence. To mitigate interference between newly observed tasks and previous tasks, we combine current-task score matching with replay over historical LoRA samples (Chaudhry et al., 2019; Song & Kingma, 2021).

We further introduce gradient alignment to encourage compatible updates between current and replay losses, and a composition consistency regularizer to smooth the local score field between task-conditioned adapter regions. This yields a continual energy learner that models LoRA parameter distributions across tasks without retraining the frozen base language model.

We widely evaluate ViT-B-32 Dosovitskiy et al. (2021), DeepSeek Bi et al. (2024), LLaVA Liu et al. (2024), and Qwen-series models Yang et al. (2025); Bai et al. (2025) on three benchmarks: the 8-task ViT benchmark, GLUE Wang et al. (2018), and MM-MergeBench Zeng et al. (2025). The results of EC-LoRA substantially outperform the data-driven continual-learning LoRA generation method under our framework. Moreover, compared with the original LoRA, EC-LoRA further optimizes it and obtains a more capable LoRA.

In conclusion, our contributions are as follows:

- We introduce **EC-LoRA**, an energy-guided implicit generative framework for generating LoRA from a learned task-conditioned energy landscape.
- We extend energy-based LoRA generation to continual generation, enabling the energy model to evolve across a sequence of tasks.
- We propose energy-guided LoRA refinement, where task prototypes computed from available LoRA checkpoints serve as stable initial points for energy-based generation.
- Experiments demonstrate that the LoRA generated by EC-LoRA outperforms those produced by data-driven approaches, while also achieving improved capabilities over the original LoRA.

2 Related Work

2.1 Continual learning

Continual learning studies how models adapt to sequentially arriving tasks while limiting performance degradation on previously learned ones McCloskey & Cohen (1989); French (1999); Parisi et al. (2019). Existing methods are commonly grouped into regularization-based, replay-based, and parameter-isolation approaches. Regularization methods, such as Elastic Weight Consolidation (EWC) Kirkpatrick et al. (2017), Synaptic Intelligence (SI) Zenke et al. (2017), and Learning without Forgetting Li & Hoiem (2017), preserve prior knowledge by constraining parameter updates or output distributions. Replay-based methods, including Experience Replay (ER) Chaudhry et al. (2019), Gradient Episodic Memory Lopez-Paz & Ranzato (2017), and Dark Experience Replay Buzzega et al. (2020), mitigate forgetting by reusing historical information during training on new tasks. Parameter-isolation methods reduce interference by allocating separate model capacity, as in Progressive Neural Networks Rusu et al. (2016) and PackNet Mallya & Lazebnik (2018). Recent work has further combined continual learning with parameter-efficient tuning. LoRA Hu et al. (2022) represents task adaptation as low-rank parameter updates, enabling task-specific knowledge to be stored with limited parameter overhead. O-LoRA Wang et al. (2023) learns orthogonal low-rank subspaces to reduce task interference, while InfLoRA Liang & Li (2024) constructs interference-free updates for continual adaptation. CL-LoRA He et al. (2025) introduces task-shared and task-specific adapters for rehearsal-free class-incremental learning. CLoRA Lu et al. (2025) constrains LoRA updates through subspace regularization, and TreeLoRA Qian et al. (2025) organizes layer-wise adapters with a gradient-similarity tree. A related but distinct line of work studies model or adapter combination, including model soups Wortsman et al. (2022), task arithmetic Ilharco et al. (2022), TIES-Merging Yadav et al. (2023), and ICM-Fusion Shao et al. (2026). Unlike these methods, EC-LoRA frames continual LoRA adaptation as energy-guided implicit parameter generation. It learns a task-conditioned energy function over LoRA parameters and produces task-specific adapters through prototype-guided refinement, rather than directly combining existing adapter weights with a fixed rule.

2.2 Energy-based Models

Energy-based models (EBMs) assign an energy value to each candidate state and perform inference by searching for low-energy configurations LeCun et al. (2006). Early EBMs were widely used for structured

prediction, where the energy function measures input-output compatibility LeCun et al. (2006); Belanger & McCallum (2016). Deep EBMs extend this principle to high-dimensional data by parameterizing the energy function with neural networks and generating samples through sampling- or optimization-based inference Du & Mordatch (2019). Because the partition function is generally intractable, score matching Hyvärinen & Dayan (2005) and denoising score matching Vincent (2011) provide practical training objectives. Related score-based generative models show that learned score fields can guide iterative refinement toward high-density regions Song & Ermon (2019). EBMs have also been used in discriminative learning, robustness, and out-of-distribution detection, where energy scores serve as confidence or compatibility measures Grathwohl et al. (2019); Liu et al. (2020). Unlike prior EBMs that typically define energy over data samples, labels, or input-output pairs, EC-LoRA defines energy over task-specific LoRA parameter vectors. LoRA generation is then carried out by refining adapter parameters toward low-energy regions of a learned task-conditioned energy landscape. These regions capture regularities shared by high-performing LoRA checkpoints and provide an inference-time mechanism for producing task-compatible adapters under continual task arrival.

3 Methodology

3.1 Problem Formulation

In this paper, we study **Continual LoRA Implicit Generation** for sequential tasks. Given a frozen foundation model $\mathcal{B}$, a sequence of tasks arrives in order as Equation 1,

$$\mathcal{T} = \{t_1, t_2, \ldots, t_T\}. \tag{1}$$

For each task t, we assume access to a set of trained LoRA checkpoints as Equation 2,

$$\mathcal{V}_t = \{v_{t,1}, v_{t,2}, \ldots, v_{t,K}\}, \quad v_{t,i} \in \mathbb{R}^D, \tag{2}$$

where $v_{t,i}$ denotes the vectorized LoRA parameters of the i-th checkpoint and D is the dimension of the full LoRA parameter space.

For a LoRA adapter $\{\Delta W_1, \Delta W_2, \ldots, \Delta W_m\}$, we first normalize each tensor as Equation 3,

$$\widehat{\Delta W_i} = \frac{\Delta W_i - \mu_i}{\sigma_i + \epsilon}, \tag{3}$$

where μ_i and σ_i are the mean and standard deviation of the i-th LoRA tensor. The normalized tensors are then flattened and concatenated as Equation 4,

$$v = \text{concat}\left(\text{vec}(\widehat{\Delta W_1}), \text{vec}(\widehat{\Delta W_2}), \ldots, \text{vec}(\widehat{\Delta W_m})\right) \in \mathbb{R}^D. \tag{4}$$

The goal is to train an energy-based generator that is continuously updated across tasks and can generate task-specific LoRA parameters for all seen tasks. Instead of using an explicit decoder $G(t) \to v$, we formulate LoRA generation as an implicit energy-guided process. Given a task condition c_t, EC-LoRA learns a conditional energy function $E_\theta(v, c_t)$, which defines a task-conditioned distribution over LoRA parameters as Equation 5,

$$p_\theta(v \mid c_t) = \frac{\exp(-E_\theta(v, c_t))}{Z_\theta(c_t)}, \quad Z_\theta(c_t) = \int \exp(-E_\theta(v, c_t))\, dv. \tag{5}$$

Under this model, lower energy corresponds to higher compatibility between a LoRA vector and the target task.

3.2 Overview

We propose **EC-LoRA**, an **Energy-Driven Continual LoRA Implicit Generation** framework. EC-LoRA learns a conditional energy landscape over the full LoRA parameter space and generates task-specific LoRA adapters by refining task prototypes toward lower-energy regions. Therefore, the overall framework consists of sequential energy learning and energy-guided inference.

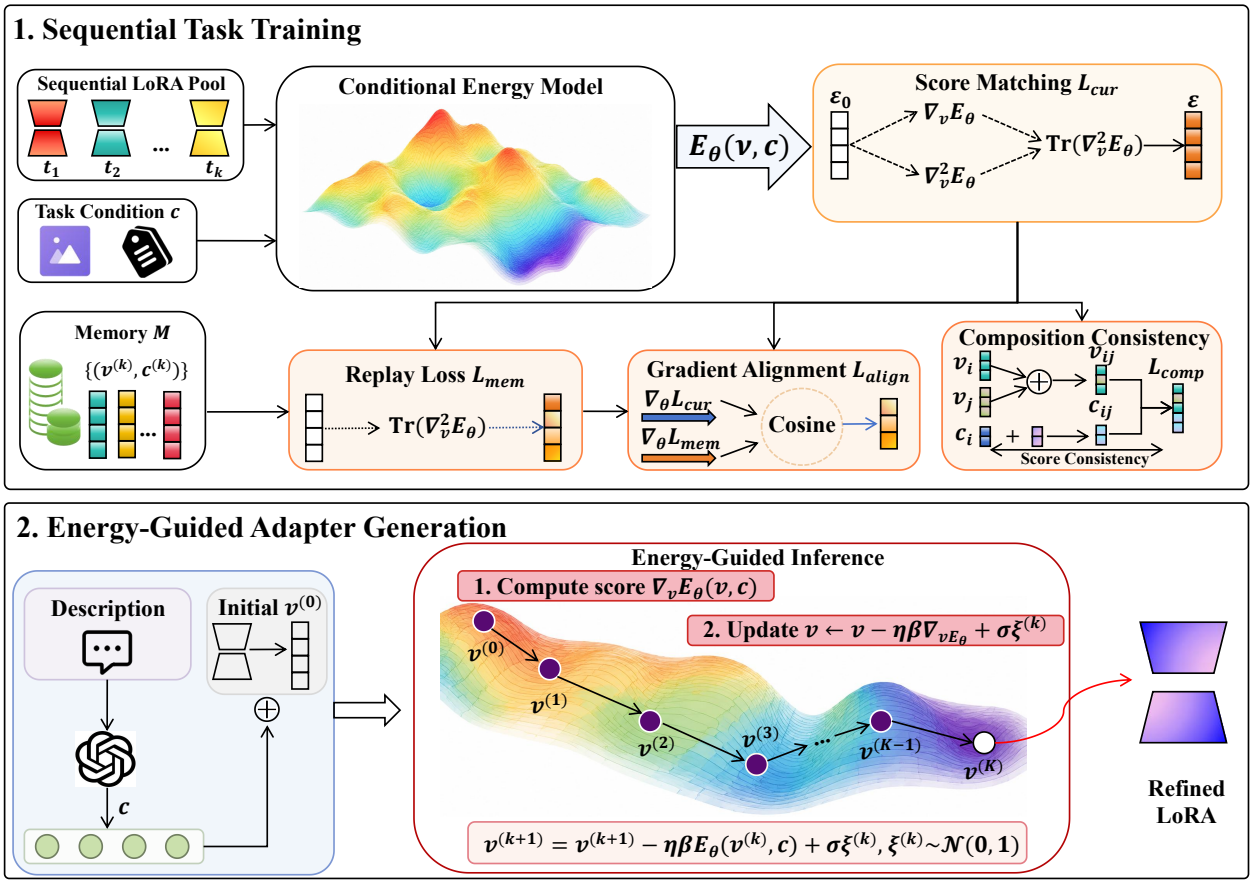


Figure 1: **Pipeline of EC-LoRA.** EC-LoRA trains a conditional EBM from sequentially observed LoRA checkpoints. Then with the guidance of the energy landscape, EC-LoRA generates the task-specific adapters by descending the learned energy landscape.

Sequential energy learning. During training, EC-LoRA uses trained LoRA checkpoints as empirical samples from the task-specific parameter regions. A task condition c_t is constructed from the task description, and the conditional energy model $E_\theta(v, c_t)$ is trained to assign low energy to effective LoRA parameters. To support continual learning, EC-LoRA maintains a reservoir memory buffer of historical LoRA vectors and introduces memory replay, gradient alignment, and composition consistency.

Energy-guided implicit generation. At inference time, EC-LoRA does not directly decode a LoRA vector in a single forward pass. Instead, it initializes generation from a task prototype $p_t = \frac{1}{K}\sum_{i=1}^{K} v_{t,i}$, and performs iterative energy descent is shown as Equation 6,

$$v_t^{(r+1)} = v_t^{(r)} - \eta\beta\nabla_v E_\theta(v_t^{(r)}, c_t). \tag{6}$$

The final refined vector is reconstructed into LoRA tensors and injected into the frozen foundational model. Thus, EC-LoRA is an implicit generator: the generated adapter is obtained by optimizing within a learned conditional energy field rather than by a direct decoder.

3.3 Conditional Energy Model

The core component of EC-LoRA is a task-conditioned energy model $E_\theta(v, c_t)$. It takes a LoRA vector v and a task condition c_t as input and outputs a scalar energy.

Task condition encoding. For each task t, we construct a task description d_t. The frozen LLM encodes d_t into token-level hidden states $H_t = \text{LLM}_{\text{base}}(d_t)$. which are summarized by a pooling operation: $h_t = \text{Pool}(H_t)$. When normalization is enabled, the task condition is defined as $c_t = \frac{h_t}{\|h_t\|_2}$.

Compared with token-based task identifiers, dense task conditions expose semantic information about the task and allow the energy model to exploit task-level similarity in the LoRA parameter space.

LoRA vector branch. Since the full LoRA vector $v \in \mathbb{R}^D$ is high-dimensional data, EC-LoRA first applies a deterministic pooling operation as Equation 7,

$$\bar{v} = \text{Pool}(v), \quad \bar{v} \in \mathbb{R}^P. \tag{7}$$

The full vector is zero-padded and divided into P consecutive chunks $\{C_j\}_{j=1}^P$. Each pooled dimension is computed as Equation 8,

$$\bar{v}_j = \frac{1}{|C_j|} \sum_{k \in C_j} v_k, \quad j = 1, \ldots, P. \tag{8}$$

The pooled vector is then encoded by a parameter branch $h_v = f_v(\bar{v})$.

Condition branch and energy head. The task condition is encoded by a condition branch $h_c = f_c(c_t)$. The parameter and condition are concatenated and passed into the energy head as $E_\theta(v, c_t) = f_E([h_v; h_c])$, where $[\cdot;\cdot]$ denotes feature concatenation. The scalar output is used as the energy assigned to v under condition c_t.

Section 3.6 gives an information-theoretic interpretation of why lower energy is expected to correspond to stronger downstream adapter performance.

3.4 Energy-Driven Continual Learning

The energy model is initialized once and updated sequentially across tasks as $\theta_{k-1} \xrightarrow{\mathcal{V}_{t_k}} \theta_k$. Unlike approaches that train a separate generator for each task, EC-LoRA maintains a single evolving conditional energy model over task-specific LoRA parameter distributions.

Score matching objective. Maximum-likelihood training is difficult because the partition function $Z_\theta(c_t)$ is intractable. Therefore, EC-LoRA uses score matching Hyvärinen & Dayan (2005) to learn the energy field. For a sample (v, c_t), the model score is $s_\theta(v, c_t) = \nabla_v E_\theta(v, c_t)$.

We adopt the Hutchinson score matching as Equation 9,

$$\mathcal{L}_{\text{SM}} = \mathbb{E}_{(v,c)} \left[\frac{1}{2} \text{mean} \left(\|\nabla_v E_\theta(v, c)\|^2 \right) - \lambda \, \text{mean} \left(\text{Tr} \left(\nabla_v^2 E_\theta(v, c) \right) \right) \right]. \tag{9}$$

The Hessian trace can be estimated with the Hutchinson estimator Hutchinson (1989) as Equation 10,

$$\text{Tr}(H) \approx \epsilon^\top H \epsilon, \quad H = \nabla_v^2 E_\theta(v, c), \tag{10}$$

where ϵ is a random probe vector sampled from a Rademacher or Gaussian distribution.

For the current task, the loss is expressed as 11,

$$\mathcal{L}_{\text{cur}} = \mathcal{L}_{\text{SM}}(v_{\text{cur}}, c_{\text{cur}}). \tag{11}$$

Reservoir memory replay. Inspired by Riemer et al., EC-LoRA maintains a reservoir memory buffer to prevent forgetting of previously learned energy landscapes as Equation 12,

$$\mathcal{M} = \{(v_i, c_i)\}_{i=1}^M. \tag{12}$$

At each update, a mini-batch of historical samples is drawn from $\mathcal{M}$, and the replay loss is computed as Equation 13,

$$\mathcal{L}_{\text{mem}} = \mathcal{L}_{\text{SM}}(v_{\text{mem}}, c_{\text{mem}}). \tag{13}$$

This term preserves low-energy regions associated with previous tasks.

Gradient alignment. Although replay preserves historical samples, current-task gradients can still conflict with memory gradients. To reduce such interference, EC-LoRA aligns the gradient directions of the current and replay losses. Let $g_{\text{cur}} = \nabla_\theta \mathcal{L}_{\text{cur}}$ and $g_{\text{mem}} = \nabla_\theta \mathcal{L}_{\text{mem}}$, then the alignment loss could be expressed as Equation 14,

$$\mathcal{L}_{\text{align}} = \max\left(0, \rho - \cos(g_{\text{cur}}, g_{\text{mem}})\right)^2, \tag{14}$$

where ρ is the target cosine similarity. This loss encourages updates for the current task to remain compatible with historical task knowledge.

Composition consistency. To stabilize the conditional energy field between task-conditioned regions, EC-LoRA introduces composition consistency. Given a current sample (v_i, c_i) and a memory sample (v_j, c_j), we construct a composed sample as Equation 15,

$$v_{ij} = \frac{1}{2}(v_i + v_j), \quad c_{ij} = \text{Normalize}(c_i + c_j). \tag{15}$$

The score at the composed point is encouraged to match the average score of the two original points as shown in Equation 16,

$$\nabla_v E_\theta(v_{ij}, c_{ij}) \approx \frac{1}{2}\left(\nabla_v E_\theta(v_i, c_i) + \nabla_v E_\theta(v_j, c_j)\right). \tag{16}$$

So the corresponding loss could be expressed as Equation 17,

$$\mathcal{L}_{\text{comp}} = \left\| \nabla_v E_\theta(v_{ij}, c_{ij}) - \frac{1}{2}\left(\nabla_v E_\theta(v_i, c_i) + \nabla_v E_\theta(v_j, c_j)\right) \right\|_2^2. \tag{17}$$

Indeed, the final continual learning objective is shown as Equation 18,

$$\mathcal{L} = \mathcal{L}_{\text{cur}} + \alpha \mathcal{L}_{\text{mem}} + \beta \mathcal{L}_{\text{align}} + \delta \mathcal{L}_{\text{comp}}, \tag{18}$$

where α, β, and δ weight memory replay, gradient alignment, and composition consistency, respectively.

3.5 Inference and LoRA Generation

After sequential training, EC-LoRA utilize the trained EBM to generate task-specific LoRA parameters. For a target task t, generation starts from the task prototype $v_t^{(0)} = p_t = \frac{1}{K}\sum_{i=1}^{K} v_{t,i}$. The prototype provides a stable initialization near the empirical high-performing region for task t.

EC-LoRA then applies R steps of energy-guided refinement as Equation 19,

$$v_t^{(r+1)} = v_t^{(r)} - \eta\beta \nabla_v E_{\theta_k}(v_t^{(r)}, c_t), \quad r = 0, \ldots, R-1, \tag{19}$$

where η is the refinement learning rate, β is the energy scaling coefficient, and θ_k denotes the EBM parameters after the k-th training stage. The final generated LoRA vector is $v_t^\star = v_t^{(R)}$.

For stochastic generation, Langevin noise can be added as Equation 20,

$$v_t^{(r+1)} = v_t^{(r)} - \eta\beta \nabla_v E_{\theta_k}(v_t^{(r)}, c_t) + \xi^{(r)}, \quad \xi^{(r)} \sim \mathcal{N}(0, \sigma^2 I). \tag{20}$$

The generated vector $v_t^\star$ is reconstructed into LoRA tensors using the template offsets and tensor shapes as Equation 21,

$$\widehat{\Delta W_i}^{\text{norm}} = \text{Unflatten}_i(v_t^\star). \tag{21}$$

Each tensor is then de-normalized with the statistics of the source checkpoint as Equation 22,

$$\widehat{\Delta W_i} = \widehat{\Delta W_i}^{\text{norm}} \cdot (\sigma_i^{\text{src}} + \epsilon) + \mu_i^{\text{src}}. \tag{22}$$

Finally, the generated adapter is loaded into the frozen base model $\mathcal{B}$ and evaluated on the target task.

3.6 Information-Theoretic Interpretation

We provide an information-theoretic justification for why lower energy indicates better downstream performance. Let $v \in \mathbb{R}^D$ denote a complete LoRA parameter vector for task t, and let τ_t be the task condition. The conditional EBM defines as Equation 23,

$$p_\psi(v \mid \tau_t) = \frac{\exp(-\Phi_\psi(v, \tau_t))}{Z_\psi(\tau_t)}, \quad Z_\psi(\tau_t) = \int \exp(-\Phi_\psi(v, \tau_t))\, dv. \tag{23}$$

Therefore,

$$-\log p_\psi(v \mid \tau_t) = \Phi_\psi(v, \tau_t) + \log Z_\psi(\tau_t). \tag{24}$$

For a fixed task t, $\log Z_\psi(\tau_t)$ is independent of v. The energy $\Phi_\psi(v, \tau_t)$ is therefore equal to the Shannon surprisal of v under the learned task-conditioned LoRA distribution, up to a task-dependent constant.

Task-risk-induced target distribution. Let the task risk of a LoRA vector v be

$$\mathcal{R}_t(v) = \mathbb{E}_{(x,y)\sim\mathcal{D}_t}\left[\ell(f_{\mathcal{B},v}(x), y)\right], \tag{25}$$

where $f_{\mathcal{B},v}$ denotes the frozen base model $\mathcal{B}$ equipped with LoRA parameter v. A natural target distribution over LoRA parameters is obtained by minimizing entropy-regularized expected risk:

$$p_t^\star = \arg\min_q \left\{\mathbb{E}_{v\sim q}\left[\mathcal{R}_t(v)\right] - \gamma\mathcal{H}(q)\right\}, \tag{26}$$

where $\mathcal{H}(q) = -\int q(v)\log q(v)\, dv$ is the Shannon entropy and $\gamma > 0$ is a temperature parameter. The first term favors low-risk LoRA parameters, while the entropy term avoids collapsing the distribution to a single point and preserves uncertainty over multiple high-quality LoRA solutions.

Using calculus of variations with the normalization constraint $\int q(v)\, dv = 1$, the Lagrangian could be expressed as Equation 26,

$$\mathcal{J}(q) = \int q(v)\mathcal{R}_t(v)\, dv + \gamma \int q(v)\log q(v)\, dv + \lambda\left(\int q(v)\, dv - 1\right). \tag{27}$$

Taking the functional derivative with respect to $q(v)$ and setting it to zero gives

$$\frac{\delta\mathcal{J}}{\delta q(v)} = \mathcal{R}_t(v) + \gamma(\log q(v) + 1) + \lambda = 0. \tag{28}$$

Hence,

$$\log q(v) = -\frac{1}{\gamma}\mathcal{R}_t(v) - \frac{\lambda+\gamma}{\gamma}, \tag{29}$$

which yields the Gibbs distribution

$$p_t^\star(v) = \frac{\exp(-\mathcal{R}_t(v)/\gamma)}{Z_t^\star}, \quad Z_t^\star = \int \exp(-\mathcal{R}_t(v)/\gamma)\, dv. \tag{30}$$

Taking the negative logarithm of Equation 30, we obtain

$$-\log p_t^\star(v) = \frac{1}{\gamma}\mathcal{R}_t(v) + \log Z_t^\star. \tag{31}$$

Thus, under the maximum-entropy principle, the surprisal of a LoRA parameter under the ideal target distribution is linearly related to its task risk.

Energy and risk equivalence under distribution calibration. The goal of EC-LoRA is to approximate the target distribution $p_t^\star(v)$ with the learned task-conditioned distribution $p_\psi(v \mid \tau_t)$. We formalize this relation through the following calibration assumption.

Assumption 1. For task t, the learned EBM is pointwise δ-calibrated with respect to the risk-induced target distribution $p_t^\star$ on the candidate LoRA parameter region $\mathcal{S}_t$, i.e.,

$$|\log p_\psi(v \mid \tau_t) - \log p_t^\star(v)| \leq \delta, \quad \forall v \in \mathcal{S}_t. \tag{32}$$

Proposition 1. Under Assumption 1, for any two LoRA vectors $v_a, v_b \in \mathcal{S}_t$, the learned energy difference approximates the risk difference:

$$|\gamma\left(\Phi_\psi(v_a, \tau_t) - \Phi_\psi(v_b, \tau_t)\right) - \left(\mathcal{R}_t(v_a) - \mathcal{R}_t(v_b)\right)| \leq 2\gamma\delta. \tag{33}$$

Proof. From the EBM definition,

$$\log p_\psi(v \mid \tau_t) = -\Phi_\psi(v, \tau_t) - \log Z_\psi(\tau_t). \tag{34}$$

Therefore, for v_a and v_b,

$$\log \frac{p_\psi(v_a \mid \tau_t)}{p_\psi(v_b \mid \tau_t)} = -\Phi_\psi(v_a, \tau_t) + \Phi_\psi(v_b, \tau_t) = -\left(\Phi_\psi(v_a, \tau_t) - \Phi_\psi(v_b, \tau_t)\right). \tag{35}$$

Similarly, from Equation 30,

$$\log \frac{p_t^\star(v_a)}{p_t^\star(v_b)} = -\frac{1}{\gamma}\mathcal{R}_t(v_a) + \frac{1}{\gamma}\mathcal{R}_t(v_b) = -\frac{1}{\gamma}\left(\mathcal{R}_t(v_a) - \mathcal{R}_t(v_b)\right). \tag{36}$$

By Assumption 1,

$$|\log p_\psi(v_a \mid \tau_t) - \log p_t^\star(v_a)| \leq \delta, \tag{37}$$

and

$$|\log p_\psi(v_b \mid \tau_t) - \log p_t^\star(v_b)| \leq \delta. \tag{38}$$

Thus,

$$\left|\log \frac{p_\psi(v_a \mid \tau_t)}{p_\psi(v_b \mid \tau_t)} - \log \frac{p_t^\star(v_a)}{p_t^\star(v_b)}\right| \leq |\log p_\psi(v_a \mid \tau_t) - \log p_t^\star(v_a)| + |\log p_\psi(v_b \mid \tau_t) - \log p_t^\star(v_b)| \leq 2\delta. \tag{39}$$

Substituting Equation 35 and Equation 36, we get

$$\left|-\left(\Phi_\psi(v_a, \tau_t) - \Phi_\psi(v_b, \tau_t)\right) + \frac{1}{\gamma}\left(\mathcal{R}_t(v_a) - \mathcal{R}_t(v_b)\right)\right| \leq 2\delta. \tag{40}$$

Multiplying both sides by γ gives

$$|\gamma\left(\Phi_\psi(v_a, \tau_t) - \Phi_\psi(v_b, \tau_t)\right) - \left(\mathcal{R}_t(v_a) - \mathcal{R}_t(v_b)\right)| \leq 2\gamma\delta. \tag{41}$$

This proves Proposition 1. □

Corollary 1. If the energy gap between two LoRA vectors satisfies

$$\Phi_\psi(v_b, \tau_t) - \Phi_\psi(v_a, \tau_t) > 2\delta, \tag{42}$$

then

$$\mathcal{R}_t(v_a) < \mathcal{R}_t(v_b). \tag{43}$$

That is, v_a has strictly lower task risk than v_b.

Proof. From Proposition 1,

$$\mathcal{R}_t(v_a) - \mathcal{R}_t(v_b) \leq \gamma \left(\Phi_\psi(v_a, \tau_t) - \Phi_\psi(v_b, \tau_t)\right) + 2\gamma\delta. \tag{44}$$

If Equation 42 holds, then

$$\Phi_\psi(v_a, \tau_t) - \Phi_\psi(v_b, \tau_t) < -2\delta. \tag{45}$$

Therefore,

$$\mathcal{R}_t(v_a) - \mathcal{R}_t(v_b) < \gamma(-2\delta) + 2\gamma\delta = 0. \tag{46}$$

Hence,

$$\mathcal{R}_t(v_a) < \mathcal{R}_t(v_b). \tag{47}$$

□

Implication for energy-based refinement. During inference, EC-LoRA refines a LoRA vector by

$$v_t^{(r+1)} = v_t^{(r)} - \eta\beta\nabla_v\Phi_\psi(v_t^{(r)}, \tau_t). \tag{48}$$

Since

$$\nabla_v \left[-\log p_\psi(v \mid \tau_t)\right] = \nabla_v\Phi_\psi(v, \tau_t), \tag{49}$$

the refinement step performs gradient descent on the surprisal of the LoRA parameter under the learned task-conditioned distribution. If one refinement step decreases the energy by

$$\Delta_r = \Phi_\psi(v_t^{(r)}, \tau_t) - \Phi_\psi(v_t^{(r+1)}, \tau_t), \tag{50}$$

then Proposition 1 implies

$$\mathcal{R}_t(v_t^{(r+1)}) - \mathcal{R}_t(v_t^{(r)}) \leq -\gamma\Delta_r + 2\gamma\delta. \tag{51}$$

Therefore, whenever

$$\Delta_r > 2\delta, \tag{52}$$

the refinement step strictly decreases task risk:

$$\mathcal{R}_t(v_t^{(r+1)}) < \mathcal{R}_t(v_t^{(r)}). \tag{53}$$

This establishes a calibrated information-theoretic connection between energy minimization and performance improvement. Lower energy means lower surprisal under the learned task-conditioned LoRA distribution; when this distribution approximates the maximum-entropy risk-induced target distribution, a sufficiently lower energy value implies lower task risk and better downstream performance in expectation.

4 Experiment

4.1 Experiment Setting

Setting Up. We evaluate EC-LoRA across visual, language, and multimodal settings to examine whether energy-guided LoRA parameter generation generalizes across model families and task types. For visual understanding tasks, we adapt our method to ViT-B-32 Dosovitskiy et al. (2021). For language understanding tasks, we adapt our method to DeepSeek-7B Bi et al. (2024) and Qwen3-8B Yang et al. (2025). For multimodal understanding tasks, we adapt our method to LLaVA-v1.5-7B Liu et al. (2024) and Qwen3-VL-8B Bai et al. (2025). All experiments were conducted on four RTX 3090 GPUs.

Baselines. We adapt data-driven continual LoRA training method to our setting, these methods include ER Chaudhry et al. (2019), CL-LoRA He et al. (2025), CLoRA Lu et al. (2025), InfLoRA Liang & Li (2024), O-LoRA Wang et al. (2023) and TreeLoRA Qian et al. (2025). We adapt our continual framework to VAE based LoRA generation method ICM-LoRA Shao et al. (2025a).

Methods	SEEN TASKS									UNSEEN TASKS				
	SciQA	Image	VQA	REC	OCR	VizWiz	Flickr	IconQA	**Average**	AVQA	Image-R	S2W	TabMWP	**Average**
Performance of MM-Merge-Bench on Qwen3VL-8B														
Zero-Shot	85.97	35.01	63.37	48.13	62.91	27.26	28.63	67.50	52.35	81.77	81.47	10.82	58.30	58.09
LoRA Hu et al. (2022)	97.71	97.74	75.22	89.21	72.83	74.94	61.69	99.16	83.56	–	–	–	–	–
ICM-LoRA Shao et al. (2025a)	93.45	88.92	67.86	81.27	62.68	60.71	57.24	87.25	74.92	84.62	67.45	13.69	71.28	59.26
ER Chaudhry et al. (2019)	80.93	74.86	60.62	76.18	64.31	55.21	46.65	78.94	67.21	75.84	58.97	9.18	60.21	51.05
CL-LoRA He et al. (2025)	81.11	32.81	52.78	76.90	54.46	42.00	43.45	71.98	56.94	67.51	60.21	8.28	56.29	48.07
CLoRA Lu et al. (2025)	82.97	30.61	54.49	76.58	58.99	42.63	51.67	75.23	59.15	67.60	69.71	12.40	55.08	51.20
InfLoRA Liang & Li (2024)	83.04	30.59	59.35	78.53	56.06	43.12	54.06	75.38	60.02	67.25	69.73	12.22	55.34	51.14
O-LoRA Wang et al. (2023)	82.94	33.71	57.64	76.61	53.01	40.27	51.52	71.21	58.36	68.68	68.61	12.37	55.17	51.21
TreeLoRA Qian et al. (2025)	83.72	32.83	52.45	76.56	52.68	41.95	47.76	72.89	57.61	67.59	60.28	8.29	55.88	48.01
EC-LoRA	**98.35**	**98.63**	**76.83**	**90.80**	**74.41**	**76.54**	**63.28**	**99.65**	**84.81**	**90.17**	**86.52**	**16.39**	**77.83**	**67.73**
Performance of MM-Merge-Bench on LLaVA														
Zero-Shot	71.73	40.87	42.88	36.10	31.23	41.03	39.07	38.09	42.62	78.45	70.36	10.05	23.10	45.49
LoRA Hu et al. (2022)	90.73	97.40	68.12	75.70	54.55	70.12	57.28	73.33	73.40	–	–	–	–	–
ICM-LoRA Shao et al. (2025a)	85.36	79.86	60.25	67.49	48.35	64.83	46.63	62.88	64.46	73.47	74.10	12.43	23.03	45.76
ER Chaudhry et al. (2019)	80.24	70.42	57.43	58.57	43.06	61.27	45.18	59.46	59.45	70.64	62.58	9.71	20.18	40.78
CL-LoRA He et al. (2025)	72.55	39.11	53.07	46.55	45.51	35.81	39.21	31.05	45.36	69.43	55.78	7.69	17.25	37.54
CLoRA Lu et al. (2025)	63.45	45.61	47.58	38.68	36.03	46.37	43.05	39.44	45.03	68.91	55.29	7.87	17.16	37.31
InfLoRA Liang & Li (2024)	65.47	42.36	48.03	48.72	32.95	36.32	42.44	49.50	45.72	68.99	55.22	7.82	17.14	37.29
O-LoRA Wang et al. (2023)	76.68	49.42	56.01	44.63	36.05	52.38	47.57	39.56	50.29	68.96	55.25	7.85	17.15	37.30
TreeLoRA Qian et al. (2025)	69.68	48.40	43.23	40.21	34.49	41.04	37.49	43.36	44.74	69.82	56.33	8.32	17.94	38.10
EC-LoRA	**92.02**	**98.40**	**69.52**	**76.93**	**58.14**	**72.78**	**57.89**	**74.25**	**74.99**	**83.95**	**78.69**	**15.04**	**27.51**	**51.30**

Table 1: **Performance of MM-Merge-Bench on eight seen and four unseen multimodal tasks.** EC-LoRA achieves strong performance on both Qwen3VL-8B and LLaVA, demonstrating its effectiveness in preserving task-specific multimodal reasoning ability and improving generalization to unseen tasks.

Datasets and Benchmarks. For visual understanding tasks, we evaluate the generated LoRA adapters under the ViT-B/32 benchmark across eight datasets of tasks, namely Cars Krause et al. (2013), MNIST LeCun (1998), EuroSAT Helber et al. (2019), GTSRB Stallkamp et al. (2011), DTD Cimpoi et al. (2014), RESISC45 Cheng et al. (2017), SUN397 Xiao et al. (2016), and SVHN Netzer et al. (2011).

For language understanding tasks, we further evaluate EC-LoRA on the GLUE benchmark Wang et al. (2018), which consists of nine English understanding tasks. These tasks cover single-sentence classification, including CoLA Warstadt et al. (2019) and SST-2; similarity and paraphrase identification, including MRPC, QQP, and STS-B; and natural language inference, including MNLI, QNLI, RTE, and WNLI.

For multimodal task, we conduct evaluations based on MM-MergeBench Zeng et al. (2025). Specifically, we assess LLAVA Liu et al. (2023) and Qwen-VL on a suite of multimodal generative tasks, including ScienceQA Lu et al. (2022a), ImageNet Deng et al. (2009), VQAv2 Goyal et al. (2017), REC-COCO Kazemzadeh et al. (2014); Mao et al. (2016), OCRVQA Mishra et al. (2019), Flickr30k Plummer et al. (2015), VizWiz-caption Gurari et al. (2018), IconQA Lu et al. (2021), ImageNet-R Hendrycks et al. (2021), AOKVQA Schwenk et al. (2022), Screen2Word Wang et al. (2021), and TabMWP Lu et al. (2022b). To examine out-of-distribution (OOD) generalization, we partition the benchmark into seen and unseen tasks. The seen tasks comprise ScienceQA, ImageNet, VQAv2, REC-COCO, OCRVQA, Flickr30k, VizWiz-caption, and IconQA, whereas the unseen tasks consist of ImageNet-R, AOKVQA, Screen2Word, and TabMWP. Although the benchmark was originally designed for model merging, we use it here to evaluate whether EC-LoRA can generate task-compatible LoRA parameters under heterogeneous multimodal task distributions.

4.2 Main Results

Results in Vision Tasks. As shown in Table 3, EC-LoRA achieves the best overall performance on the ViT-B/32 visual recognition benchmark. Compared with the zero-shot model, EC-LoRA substantially improves performance across diverse visual tasks, showing that the generated LoRA adapters recover task-specific visual discrimination ability. Although single-task LoRA is a strong baseline, it requires maintaining a separate adapter for each task, whereas EC-LoRA produces task-specific adapters through a unified energy-guided generation framework. Compared with ICM-LoRA and continual LoRA baselines, EC-LoRA obtains more consistent improvements across datasets, indicating that fixed adapter combination rules or purely sequential LoRA updates are less effective when task distributions vary substantially.

Methods	SST-2	MRPC	RTE	QNLI	QQP	CoLA	MNLI-m	MNLI-mm	STS-B	WNLI	Avg.
GLUE benchmark results on DeepSeek											
Original Model	60.89	58.03	47.65	52.11	40.13	7.85	33.70	34.02	13.87	47.88	39.61
LoRA Hu et al. (2022)	96.56	88.04	88.45	94.76	90.03	65.97	90.92	90.84	90.14	77.46	87.32
ICM-LoRA Shao et al. (2025a)	93.70	84.57	85.75	92.64	85.84	63.84	86.03	85.81	88.57	74.64	84.14
ER Chaudhry et al. (2019)	67.29	57.81	69.43	77.65	55.23	24.18	48.18	50.19	59.15	57.74	56.69
CL-LoRA He et al. (2025)	49.08	38.26	38.62	46.27	67.63	11.83	33.81	33.88	43.18	66.19	42.88
CLoRA Lu et al. (2025)	54.86	72.97	31.05	43.27	60.03	17.88	30.29	30.59	54.75	57.74	45.34
InfLoRA Liang & Li (2024)	58.72	72.33	32.49	43.90	61.07	31.94	52.31	52.81	54.50	74.64	53.47
O-LoRA Wang et al. (2023)	53.66	72.55	31.04	43.14	60.04	40.57	50.07	50.38	54.76	73.23	52.94
TreeLoRA Qian et al. (2025)	67.32	61.13	37.90	45.41	67.25	35.92	44.30	44.66	60.12	67.60	53.16
EC-LoRA	**97.36**	**88.91**	**89.72**	**95.32**	**91.42**	**67.99**	**91.26**	**91.09**	**90.63**	**78.87**	**88.26**
GLUE benchmark results on Qwen											
Original Model	74.03	64.61	66.64	70.48	51.83	42.89	51.02	49.96	64.45	39.43	57.53
LoRA Hu et al. (2022)	96.65	90.25	89.16	95.62	90.39	67.42	91.12	91.26	90.05	70.42	87.23
ICM-LoRA Shao et al. (2025a)	92.56	82.25	80.78	87.38	83.89	58.61	80.73	81.66	86.82	71.83	80.65
ER Chaudhry et al. (2019)	71.25	74.18	75.94	71.73	68.70	48.52	56.90	56.95	67.23	47.60	63.90
CL-LoRA He et al. (2025)	50.22	54.77	46.93	50.59	46.67	21.79	34.36	35.80	53.57	64.78	45.95
CLoRA Lu et al. (2025)	51.83	75.50	39.35	53.78	47.16	27.45	36.60	37.95	58.72	64.78	49.31
InfLoRA Liang & Li (2024)	50.91	79.82	40.07	50.59	47.22	28.78	41.64	42.53	46.96	60.56	48.91
O-LoRA Wang et al. (2023)	51.60	71.47	39.71	50.55	47.19	29.39	45.77	46.84	58.77	63.38	50.47
TreeLoRA Qian et al. (2025)	50.34	74.77	45.48	55.41	46.83	41.38	51.74	52.41	57.30	64.78	54.04
EC-LoRA	**97.06**	**90.88**	**90.85**	**95.94**	**91.17**	**67.04**	**91.82**	**91.78**	**90.23**	**71.83**	**87.86**

Table 2: **GLUE benchmark results on DeepSeek and Qwen.** EC-LoRA achieves strong performance across diverse language understanding tasks on both backbones, demonstrating its effectiveness in preserving task-specific utility and improving cross-task transfer.

Method	MNIST	GTSRB	EuroSAT	RESISC45	DTD	SUN397	Stanford Cars	SVHN	Avg.
Zero-Shot	51.77	43.47	54.11	66.36	44.91	65.54	64.63	51.98	55.35
LoRA Hu et al. (2022)	99.78	99.13	98.03	96.60	80.02	77.16	89.47	97.30	92.19
ICM-LoRA Shao et al. (2025a)	94.46	96.89	92.25	94.37	74.55	71.82	82.14	89.25	86.97
ER Chaudhry et al. (2019)	87.89	82.66	79.13	80.35	65.50	67.87	78.53	85.85	78.47
CL-LoRA Lu et al. (2025)	45.37	51.81	42.88	32.98	42.18	58.25	67.73	64.79	50.75
CLoRA Lu et al. (2025)	48.31	47.19	51.44	44.04	53.86	43.29	53.46	67.81	51.17
InfLoRA Liang & Li (2024)	54.53	55.42	51.85	76.19	61.70	27.76	43.52	76.48	55.93
O-LoRA Wang et al. (2023)	47.95	33.61	53.80	48.39	46.71	37.03	37.77	65.31	46.32
TreeLoRA Qian et al. (2025)	58.32	52.22	48.09	53.19	48.25	17.57	58.83	74.58	51.38
EC-LoRA	**99.79**	**99.49**	**98.68**	**96.79**	**80.32**	**77.51**	**89.65**	**97.70**	**92.49**

Table 3: **Results on eight visual recognition tasks with CLIP-ViT-B/32 as the pretrained foundation model.** EC-LoRA consistently improves the average accuracy over competitive continual LoRA and replay-based baselines, showing stronger robustness in visual recognition of generated LoRA.

The advantage of EC-LoRA comes from explicitly modeling the task-conditioned LoRA parameter space. Existing adapter combination methods usually rely on fixed parameter-space operations and do not model the geometry of high-performing adapter regions. Continual learning methods mainly preserve previous knowledge through replay or update constraints, but they do not directly optimize the generated adapter at inference time. EC-LoRA addresses this limitation by learning a task-conditioned energy landscape and refining task prototypes toward low-energy regions. This allows the generated adapter to retain useful structure from high-performing checkpoints while becoming more compatible with the target visual task.

Results in Multimodal Tasks. As shown in Table 1, EC-LoRA achieves consistently strong performance on multimodal tasks across both Qwen3-VL-8B and LLaVA-V1.5-7B. Compared with the original models, EC-LoRA substantially improves overall performance, demonstrating that the generated adapters can enhance multimodal reasoning, visual understanding, OCR recognition, and image-text alignment. Compared with ICM-LoRA, EC-LoRA achieves better results on both seen and unseen tasks, suggesting that regression-

style parameter prediction or fixed adapter combination may be insufficient for heterogeneous multimodal task distributions. EC-LoRA instead learns a conditional energy model over the LoRA parameter space and generates adapters by refining task prototypes, providing a more reliable mechanism for producing task-compatible LoRA parameters.

More importantly, EC-LoRA shows strong generalization in the unseen-task setting. Since unseen tasks provide no target-domain LoRA adapters during evaluation, methods that depend on memorizing or combining task-specific adapters are not matched to this scenario. The strong performance of EC-LoRA indicates that the learned energy landscape captures transferable structure among seen-task LoRA parameters and can guide adapter generation for new task distributions. Compared with continual learning baselines, which mainly reduce forgetting through replay or parameter constraints, EC-LoRA directly refines the generated adapter toward a more compatible region in the LoRA parameter space. This helps reduce negative transfer from irrelevant tasks while preserving useful cross-task structure, leading to stronger out-of-distribution multimodal generalization.

Results in Language Tasks. As shown in Table 2, EC-LoRA achieves the best overall performance on GLUE across both DeepSeek and Qwen backbones. Compared with the original models, EC-LoRA brings clear improvements on language understanding tasks with different forms of reasoning, including sentiment classification, paraphrase detection, natural language inference, semantic similarity, and linguistic acceptability. Compared with ICM-LoRA and continual LoRA baselines, EC-LoRA shows more stable performance across tasks, indicating that its generated adapters better preserve task-specific language knowledge.

For language tasks, conflicting optimization directions across tasks can make fixed adapter combination or sequential updating produce suboptimal LoRA parameters. Continual learning baselines primarily suppress forgetting, but they do not explicitly improve the generated adapter for a target task. EC-LoRA alleviates this issue by learning a task-conditioned energy function over the normalized LoRA parameter space and refining prototypes toward regions associated with high-quality task adaptation. Moreover, EC-LoRA generates only LoRA tensors and reuses the non-LoRA task head from the selected source checkpoint, making it suitable for language tasks with heterogeneous output spaces. Together with the vision and multimodal results, these findings demonstrate that EC-LoRA is a task-agnostic and modality-general framework for generating high-quality LoRA adapters under continual task arrival.

4.3 Ablation Study

All ablation experiments are conducted using ViT-B/32 on the eight-task ViT-B/32 benchmark.

Effect of Major Components. Table 4 provides a direct overview of how each component contributes to EC-LoRA. The full method achieves the best final average performance and the lowest forgetting, showing that the proposed framework benefits from the joint design of task conditioning, prototype-guided refinement, and continual energy learning. The prototype-only variant already performs competitively, indicating that averaging high-performing LoRA checkpoints provides a meaningful task-level initialization. However, its performance remains lower than the full method, which demonstrates that prototype construction alone cannot fully recover the optimal task-specific adapter and still requires energy-based refinement.

Method	Mean Avg. ↑	Final Avg. ↑	Forgetting↓	BWT ↑
Source Adapter	–	92.19	–	–
Prototype Only	90.45	91.27	0.31	-0.02
w/o Task Condition	88.74	89.36	0.49	-0.04
w/o Replay	85.67	85.74	1.43	-0.19
w/o Align	91.08	91.91	0.30	-0.02
w/o Comp	91.19	92.03	0.29	-0.02
w/o All CL Modules	83.92	84.31	1.87	-0.26
Full Method	**91.63**	**92.49**	**0.25**	**-0.01**

Table 4: **Effect of Major component.** The full method obtains the best final average and the lowest forgetting, confirming the importance of task conditioning, replay, gradient alignment, and composition consistency for continual LoRA generation.

Removing the task condition clearly weakens performance, confirming that EC-LoRA needs task-aware guidance rather than a task-agnostic energy function. Among the continual learning components, removing

Steps	Mean Avg. ↑	Final Avg. ↑	Energy Drop ↑	Refined Energy ↓	Init-Refined Cosine ↑	Relative Shift L2 ↓	Sample Loss ↓	Shift L2	Shift RMS	Shift Max Abs
0	90.45	91.27	–	–	–	–	0.68	–	–	–
1	90.96	91.81	0.42	-0.23	**0.99**	**0.02**	0.62	4.33	0.00	0.04
3	91.38	92.27	1.08	-0.90	0.98	0.04	0.57	9.85	0.00	0.09
5	91.63	92.49	1.44	-1.25	0.97	0.06	0.55	13.58	0.01	0.12
10	**91.85**	**92.56**	**1.90**	**-1.51**	0.98	0.08	**0.54**	21.73	0.01	0.21

Table 5: **Ablation study on the number of refinement steps.** Increasing the number of energy-based refinement steps improves the generated adapter quality, showing that iterative energy minimization effectively guides LoRA parameters toward better-performing regions.

Learning Rate	Mean Avg. ↑	Final Avg. ↑	Energy Drop ↑	Refined Energy ↓	Init-Refined Cosine ↑	Relative Shift L2 ↓	Sample Loss ↓	Shift L2	Shift RMS	Shift Max Abs
1×10^{-5}	90.83	91.64	0.32	-0.13	**1.00**	**0.01**	0.63	3.10	0.00	0.03
3×10^{-5}	91.29	92.13	0.87	-0.68	0.99	0.04	0.59	8.22	0.00	0.08
1×10^{-4}	**91.63**	**92.49**	1.44	-1.25	0.97	0.06	**0.55**	13.58	0.01	0.12
3×10^{-4}	91.07	91.86	**2.18**	**-2.00**	0.93	0.13	0.60	28.94	0.01	0.29

Table 6: **Ablation study on the refinement learning rate.** A moderate learning rate achieves the best balance between energy reduction and parameter stability, indicating that overly small or large refinement updates can limit final performance.

replay causes the most severe degradation, while removing gradient alignment or composition consistency results in milder but consistent drops. This suggests that replay is the foundation for preserving historical task regions, whereas alignment and composition consistency further improve the stability of energy learning. The variant without all continual learning modules performs the worst, verifying that EC-LoRA relies on both adapter generation and continual energy regularization to handle sequential task arrival.

Effect of Energy-Based Refinement. Refinement is the step that transforms a task prototype into the final generated adapter. Table 5 shows that increasing the number of refinement steps improves downstream performance and reduces refined energy, confirming that the learned energy function provides a meaningful optimization direction. The zero-step setting corresponds to using the prototype without refinement, which performs worse than refined variants. This indicates that the prototype provides a reliable initialization, while energy descent further improves adapter quality by moving the LoRA parameters toward a more compatible region.

Tables 6 and 7 show that this refinement process must be properly controlled. A small learning rate or energy scaling coefficient provides insufficient refinement and limits the improvement over the prototype. In contrast, overly large values move the generated parameters more aggressively, which may weaken stability even when the energy decreases more strongly. Therefore, EC-LoRA adopts a moderate refinement strategy to balance energy-guided improvement and parameter-space stability. These results demonstrate that the benefit of refinement comes from controlled task-aware correction rather than simply increasing the magnitude of parameter updates.

Effect of Prototype Initialization and Task Condition. The quality of generated adapters depends on both the starting point of refinement and the condition used to guide the energy model. Table 8 shows that constructing the prototype from a compact set of high-performing checkpoints gives the best overall result. Using too few checkpoints makes the prototype sensitive to individual LoRA solu-

Top-K	Mean Avg. ↑	Final Avg. ↑	Forgetting↓	Prototype Norm	RefinementΔ
1	91.54	92.44	0.27	281.44	12.95
5	91.59	92.45	0.26	278.59	13.22
10	**91.63**	**92.49**	**0.25**	**276.83**	**13.58**
50	91.47	92.33	0.29	268.72	14.37

Table 8: **Ablation study on the number of top-K checkpoints.** Using the top-10 checkpoints produces the strongest overall performance, suggesting that a compact set of high-quality LoRA samples provides a reliable prototype for energy-based generation.

Beta	Mean Avg. ↑	Final Avg. ↑	Energy Drop ↑	Refined Energy ↓	Init-Refined Cosine ↑	Relative Shift L2 ↓	Sample Loss ↓	Shift L2	Shift RMS	Shift Max Abs
0.5	91.21	92.07	0.76	-0.58	**0.99**	**0.03**	0.59	7.85	0.00	0.07
1.0	**91.63**	**92.49**	1.44	-1.25	0.97	0.06	**0.55**	13.58	0.01	0.12
2.0	91.39	92.24	**2.10**	**-1.91**	0.95	0.11	0.57	24.32	0.01	0.24

Table 7: **Effects of the refinement beta.** The results show that an intermediate energy scaling coefficient provides the most effective refinement, balancing sufficient energy descent with stable parameter movement.

Condition	Mean Avg. ↑	Final Avg. ↑	Forgetting↓	**Comment**
one-hot	88.74	89.36	0.49	task identity only
dense-hidden, no norm	91.17	92.08	0.31	semantic condition without normalization
dense-hidden, last pooling	91.02	91.89	0.34	final-token representation
dense-hidden, mean pooling	**91.63**	**92.49**	**0.25**	**full method**

Table 9: **Task condition ablation study of EC-LoRA.** Dense hidden task representations with mean pooling achieve the best performance, demonstrating that semantic task conditioning is more effective than using task identity alone.

tions, while using too many checkpoints may introduce less representative adapters and reduce task specificity. This suggests that the prototype should capture stable high-quality task information instead of averaging all available LoRA samples indiscriminately.

Table 9 further confirms the importance of task-conditioned generation. Dense task representations outperform one-hot task identifiers, showing that semantic task information is more effective than task identity alone. One-hot conditions can distinguish different tasks, but they cannot describe task similarity or provide semantic guidance for adapter generation. In contrast, dense hidden representations extracted from task descriptions allow the energy model to capture relationships among tasks. Mean pooling with normalization achieves the strongest result, indicating that stable task-level semantic representations are important for guiding LoRA parameters toward task-compatible regions.

Effect of Continual Energy Learning. After identifying the importance of continual learning components, Tables 10, 11, and 12 further examine how historical information should be preserved. The module-combination results show that replay, gradient alignment, and composition consistency play complementary roles. Replay maintains the LoRA samples and low-energy regions associated with previous tasks. Gradient alignment reduces optimization conflicts between current-task and replay gradients. Composition consistency encourages smoother transitions between different task-conditioned regions. Using alignment or composition consistency alone is insufficient, because these regularizers cannot preserve old-task distributions without historical samples.

Memory Size	Avg. Score ↑	Final Avg. ↑	Forgetting↓	Old-task Avg. ↑	New-task Avg. ↑
0	85.67	85.74	1.43	84.61	**92.83**
1	88.73	88.71	0.93	88.12	92.76
2	90.14	90.97	0.54	90.54	92.68
3	90.82	91.84	0.36	91.43	92.57
4	91.63	92.49	0.25	92.26	92.41
8	**91.71**	**92.55**	**0.23**	**92.40**	92.33

Table 11: **Ablation study on replay memory size.** Larger replay memory improves old-task retention and final average performance, showing that historical LoRA samples are important for preserving the learned energy landscape.

The memory and replay sampling results further show that EC-LoRA requires not only replay, but also informative replay. Increasing the memory size generally improves old-task retention, indicating that broader coverage of historical LoRA helps maintain the learned energy landscape. However, replay sampling must balance relevance and diversity. Random sampling provides weaker guidance, while cosine-based sampling

CL Modules	Mean Avg. ↑	Final Avg. ↑	Forgetting↓	BWT ↑	**Comment**
None	83.92	84.31	1.87	-0.26	w/o all CL modules
Replay only	90.54	91.21	0.43	-0.05	replay without auxiliary constraints
Align only	84.76	85.19	1.64	-0.23	align without replay and comp
Comp only	84.31	84.87	1.70	-0.25	comp without replay and align
Replay + Align	91.19	92.03	0.29	-0.02	w/o comp
Replay + Comp	91.08	91.91	0.30	-0.02	w/o align
Align + Comp	85.67	85.74	1.43	-0.19	w/o replay
Replay + Align + Comp	**91.63**	**92.49**	**0.25**	**-0.01**	**full method**

Table 10: **Ablation study on continual learning module combinations.** Combining replay, gradient alignment, and composition consistency yields the best final average and lowest forgetting, confirming their complementary roles in mitigating continual learning interference.

Replay Sampling Mode	**Top-K**	Avg. Score ↑	Final Avg. ↑	Forgetting↓	Old-task Avg. ↑	New-task Avg. ↑	Replay Similarity ↑
random	0	90.74	91.35	0.41	90.91	**92.64**	0.21
cosine_topk2	2	91.42	92.27	0.28	92.04	92.52	**0.68**
cosine_topk4	**4**	**91.63**	**92.49**	**0.25**	**92.26**	92.41	0.59
cosine_topk8	8	91.57	92.43	0.27	92.19	92.38	0.44
cosine_topk10	10	91.49	92.35	0.29	92.08	92.36	0.35

Table 12: **Ablation study on replay sampling strategy.** Cosine-based replay sampling with top-4 neighbors achieves the strongest balance between replay relevance and diversity, leading to the best overall continual learning performance.

with a moderate number of neighbors better selects historical samples related to the current task without collapsing to an overly narrow subset. These results show that EC-LoRA preserves continual knowledge by maintaining structured historical regions in the energy landscape, rather than storing previous adapters.

Effects of Energy Model Design. The last group of ablations studies whether the learned energy landscape is sufficiently expressive and well trained. Table 13 shows that moderate EBM capacity achieves the best trade-off between performance and training cost. When the pooled vector length or hidden dimension is too small, the energy model cannot adequately capture the structure of high-quality LoRA. Increasing the capacity improves expressiveness, but excessive capacity does not bring further gains and increases computational cost. This suggests that EC-LoRA benefits from an energy model that is expressive enough to represent task-conditioned adapter regions while remaining compact enough for stable continual learning.

Table 14 shows that the score matching objective directly affects the quality of energy-based generation. Hutchinson score matching with a moderate trace coefficient achieves the strongest performance, indicating that the proposed objective learns a useful score field for guiding LoRA refinement. Weak trace regularization provides insufficient constraint on the local energy geometry, while overly strong regularization may over-smooth the energy landscape and reduce task-specific refinement ability. The denoising score matching alternative performs worse than the full objective, further confirming that the proposed training formulation is better suited for prototype-guided energy-based LoRA generation.

5 Analyses

Since EC-LoRA performs energy-based implicit generation, an important question is whether the learned energy is aligned with downstream adapter quality. Our theoretical analysis suggests that, if the learned EBM is calibrated with the risk-induced Gibbs distribution, the energy gap between two LoRA parameters should approximate their task-risk gap:

$$\gamma\left(\Phi_\psi(v_a,\tau_t)-\Phi_\psi(v_b,\tau_t)\right)\approx R_t(v_a)-R_t(v_b). \tag{54}$$

Setting	**Value**	Avg. Score ↑	Final Avg. ↑	Forgetting↓	Energy Drop ↑	RefinementΔ	Train Time ↓
Pool length	512	90.86	91.72	0.36	0.94	9.68	**0.72**
Pool length	1024	91.28	92.18	0.29	1.19	11.84	0.86
Pool length	2048	**91.63**	**92.49**	**0.25**	1.44	13.58	1.00
Pool length	4096	91.58	92.43	0.27	**1.51**	14.96	1.37
Vector hidden dim	128	91.12	92.04	0.32	1.10	11.27	**0.81**
Vector hidden dim	192	**91.63**	**92.49**	**0.25**	1.44	13.58	1.00
Vector hidden dim	384	91.54	92.38	0.28	**1.68**	16.93	1.58
Condition hidden dim	64	91.18	92.09	0.31	1.16	11.92	**0.89**
Condition hidden dim	128	**91.63**	**92.49**	**0.25**	1.44	13.58	1.00
Condition hidden dim	256	91.51	92.36	0.28	**1.57**	15.35	1.31

Table 13: **Ablation study on EBM capacity.** Moderate EBM capacity achieves the best trade-off between performance and training cost, indicating that sufficient but not excessive model capacity is needed for stable LoRA energy modeling.

Objective	Trace / Noise	Avg. Score ↑	Final Avg. ↑	Forgetting ↓	Score Loss ↓	Energy Drop ↑	Refinement Δ	Sample Loss ↓	**Comment**
Hutchinson	trace=0.05	91.21	92.08	0.31	0.02	1.09	10.75	0.59	weak trace regularization
Hutchinson	trace=0.1	91.48	92.31	0.28	0.02	1.28	12.16	0.56	moderate trace regularization
Hutchinson	trace=0.5	91.12	91.93	0.34	0.03	**1.69**	16.44	0.60	overly strong trace regularization
Denoising	noise=0.1	90.94	91.76	0.37	0.03	1.21	14.03	0.61	DSM alternative
Hutchinson	trace=0.2	**91.63**	**92.49**	**0.25**	**0.02**	1.44	13.58	**0.55**	**full method**

Table 14: **Ablation study on the score matching objective.** Hutchinson score matching with a moderate trace coefficient achieves the strongest performance, validating the effectiveness of the proposed objective for learning a useful LoRA energy landscape.

This implies that lower-energy LoRA parameters should correspond to lower validation risk and stronger task performance. We therefore analyze the learned energy from four perspectives: energy–performance alignment, pairwise energy–risk calibration, refinement trajectory, and finite-candidate Gibbs calibration.

Energy and Performance Alignment. We first examine whether low-energy LoRA parameters tend to achieve better validation behavior. For each task (t), we collect candidate LoRA vectors from original high-performing checkpoints, task prototypes, intermediate refinement states, and final generated adapters. For each candidate $(v_{t,i})$, we compute its learned energy, validation risk, and task metric:

$$E_{t,i} = \Phi_\psi(v_{t,i}, \tau_t), \quad R_{t,i} = \mathcal{L} * \mathrm{val}(v * t, i), \quad A_{t,i} = \mathrm{Metric}(v_{t,i}). \tag{55}$$

Figure 2(a) groups candidates by refined-energy quantiles and reports the corresponding task-normalized accuracy gain. Candidates in lower-energy quantiles generally obtain positive accuracy gains, while high-energy candidates tend to show weaker or negative gains. Figure 2(b) shows the same trend from the validation-risk perspective. Lower-energy candidates obtain better validation-loss responses, whereas higher-energy candidates are more likely to increase validation loss. These results show that the learned energy is an effective surrogate for identifying high-quality LoRA regions: lower energy usually indicates lower validation risk and better downstream performance.

Pairwise Energy–Risk Calibration. The previous analysis evaluates candidates individually. We further study whether the relative ordering induced by energy is consistent with the relative ordering induced by validation risk. For two candidates (v_a) and (v_b) from the same task, we define

$$\Delta E_{ab} = \Phi_\psi(v_a, \tau_t) - \Phi_\psi(v_b, \tau_t), \tag{56}$$

$$\Delta R_{ab} = R_t(v_a) - R_t(v_b). \tag{57}$$

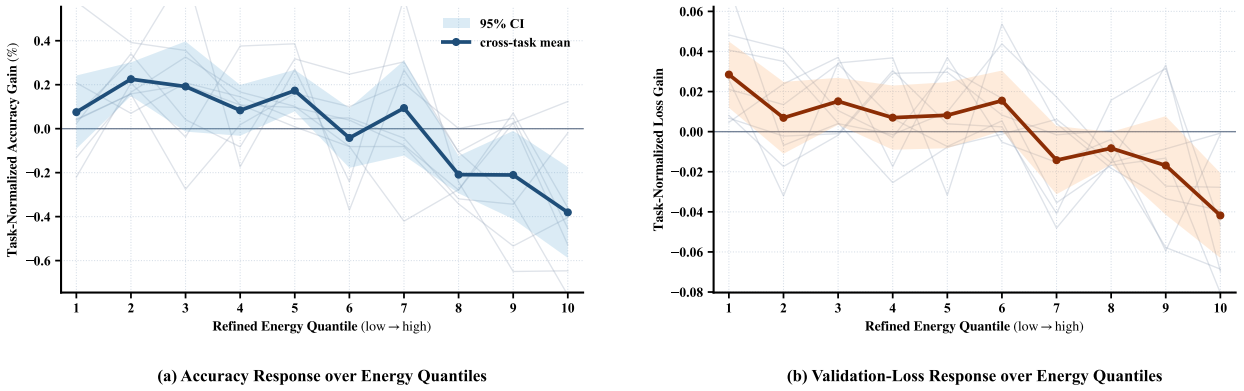


Figure 2: **Energy and performance alignment of EC-LoRA.** We group candidate LoRA parameters by refined-energy quantiles and compare their downstream behavior. (a) Lower-energy candidates generally yield higher task-normalized accuracy gains. (b) Lower-energy candidates also obtain better validation-loss responses. These results show that the learned energy provides a useful surrogate for identifying high-quality LoRA regions.

Figure 3 visualizes the relation between (ΔE_{ab}) and (ΔR_{ab}). The conditional mean of the pairwise risk gap increases as the pairwise energy gap becomes larger, indicating that candidates with relatively lower energy also tend to have relatively lower validation risk. This directly supports the theoretical prediction that energy gaps can approximate risk gaps. Although individual pairs may deviate from the trend due to finite validation samples and task-specific noise, the overall monotonic relation shows that the EBM learns a meaningful ordering over candidate LoRA parameters. Therefore, EC-LoRA can use energy descent as a practical inference-time mechanism for improving adapter quality.

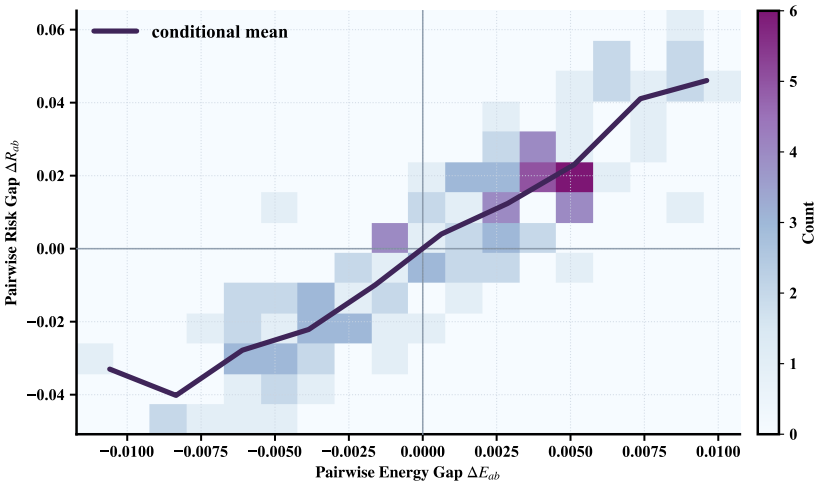
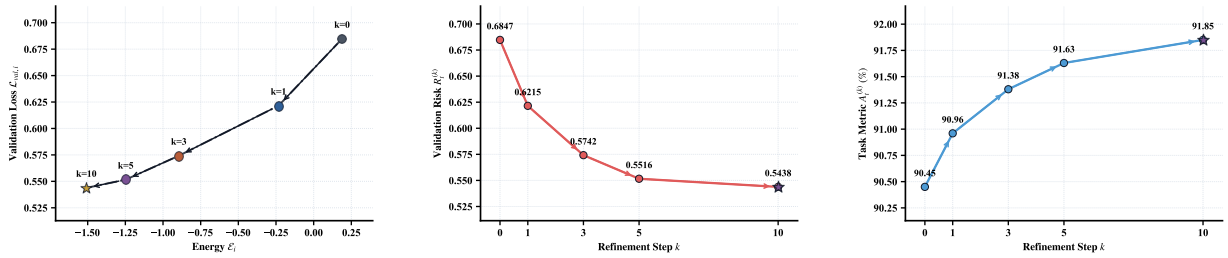


Figure 3: **Pairwise energy and risk calibration.** For each task, we compare the pairwise energy gap (ΔE_{ab}) with the pairwise validation-risk gap (ΔR_{ab}) between candidate LoRA parameters. The increasing conditional mean indicates that lower relative energy generally corresponds to lower relative risk, supporting the theoretical claim that calibrated energy gaps approximate risk gaps.

Refinement Trajectory. We next analyze whether the refinement process actually moves the generated adapter toward lower-risk regions. Starting from the task prototype $v_t^{(0)} = p_t$, we record the refinement trajectory $(v_t^{(0)}, v_t^{(1)}, \ldots, v_t^{(K)})$. At each refinement step, we compute

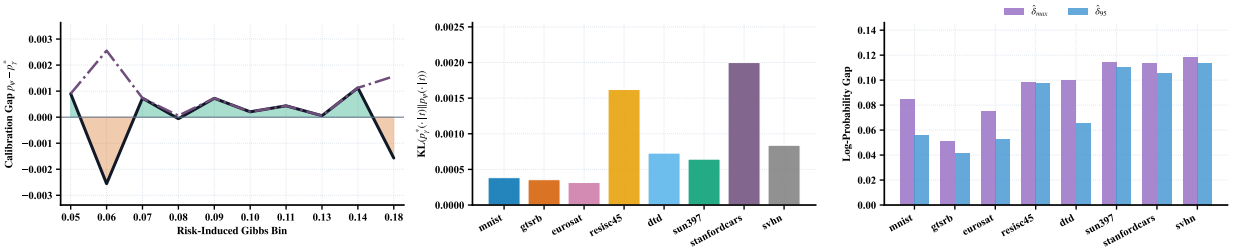
$$E_t^{(k)} = \Phi_\psi(v_t^{(k)}, \tau_t), \quad R_t^{(k)} = R_t(v_t^{(k)}), \quad A_t^{(k)} = \mathrm{Metric}(v_t^{(k)}). \tag{58}$$

Figure 4(a) shows the trajectory in the energy–validation-loss space. As refinement proceeds, the generated adapter moves toward lower-energy and lower-risk regions. Figure 4(b) further shows that validation risk decreases along the refinement steps, while Figure 4(c) shows that the task metric improves accordingly. These results confirm that energy-based refinement is not merely a procedural update after prototype construction. Instead, it provides an effective optimization path that gradually converts a stable task prototype into a stronger task-specific adapter.



(a) Refinement Trajectory (b) Validation Risk vs. Refinement Step (c) Task Metric vs. Refinement Step

Figure 4: **Energy-based refinement trajectory.** Starting from the task prototype, EC-LoRA iteratively refines the generated LoRA parameters under the learned energy model. (a) The refinement trajectory moves toward lower-energy and lower-risk regions. (b) Validation risk decreases along refinement steps. (c) The task metric improves as refinement proceeds. These results show that prototype-guided energy descent provides an effective path for improving generated adapter quality.



(a) Finite-Candidate Gibbs Calibration Gap (b) KL Calibration by Task (c) Log-Probability Gap by Task

Figure 5: **Finite-candidate Gibbs calibration.** We compare the learned EBM distribution with the validation-risk-induced Gibbs distribution over a finite set of candidate LoRA parameters. (a) The calibration gap remains close to zero across candidate bins. (b) The task-level KL divergence remains small across tasks. (c) The maximum and high-percentile log-probability gaps remain bounded. These results indicate that the learned EBM distribution is well aligned with the risk-induced Gibbs distribution.

The improvement also becomes more gradual after several refinement steps, suggesting that EC-LoRA mainly benefits from controlled local refinement rather than excessive parameter movement. This observation is consistent with the ablation results on refinement steps and refinement hyperparameters. A moderate refinement process can reduce learned surprisal and validation risk, while maintaining the useful structure of the prototype.

Finite-Candidate Gibbs Calibration. Finally, we evaluate whether the learned EBM distribution is calibrated with the risk-induced Gibbs distribution over a finite candidate set. For each task (t), given a finite candidate set

$$\mathcal{S} * t = v * t, 1, \ldots, v_{t,N}, \tag{59}$$

we define the learned EBM distribution as

$$p_\psi(i \mid t) = \frac{\exp(-E_{t,i})}{\sum_{j=1}^{N} \exp(-E_{t,j})}, \tag{60}$$

and the risk-induced Gibbs distribution as

$$p^* * \gamma(i \mid t) = \frac{\exp(-R * t, i/\gamma)}{\sum_{j=1}^{N} \exp(-R_{t,j}/\gamma)}. \tag{61}$$

Figure 5(a) compares the learned distribution with the risk-induced distribution across candidate probability bins. The calibration gap stays close to zero, showing that the EBM assigns probability mass to candidate regions in a way that is consistent with the validation-risk-induced Gibbs distribution. Figure 5(b) reports the task-level KL calibration error. The consistently small KL values indicate that this calibration behavior is stable across tasks rather than being dominated by a single dataset. Figure 5(c) further reports the maximum and high-percentile log-probability gaps, which remain bounded across tasks. This suggests that the learned EBM does not severely distort the likelihood of most candidate adapters.

Overall, these analyses empirically support the information-theoretic interpretation of EC-LoRA. The learned energy is aligned with validation risk, pairwise energy gaps reflect pairwise risk gaps, refinement decreases both energy and validation risk, and the induced finite-candidate distribution is close to the risk-induced Gibbs distribution. These observations explain why EC-LoRA can improve adapter generation by minimizing a learned task-conditioned energy function: under calibrated EBM learning, lower energy corresponds to lower task risk and better downstream performance.

6 Conclusion

We presented EC-LoRA, an energy-driven continual LoRA implicit generation framework for task-specific adapter generation under sequential task arrival. Instead of directly decoding LoRA weights, EC-LoRA learns a task-conditioned energy function over the normalized LoRA parameter space and generates adapters through prototype-guided energy refinement. To mitigate forgetting, EC-LoRA preserves historical LoRA distributions with memory replay, gradient alignment, and composition consistency. Experiments demonstrate that EC-LoRA consistently improves generated adapter quality, maintains strong performance on previously observed tasks, and achieves favorable generalization to unseen task settings. Future work will explore more scalable energy-based LoRA generation with reduced dependence on stored checkpoint pools.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

David Belanger and Andrew McCallum. Structured prediction energy networks. In *International Conference on Machine Learning*, pp. 983–992. PMLR, 2016.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in neural information processing systems*, 32, 2019.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4): 128–135, 1999.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3608–3617, 2018.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Jiangpeng He, Zhihao Duan, and Fengqing Zhu. Cl-lora: Continual low-rank adaptation for rehearsal-free class-incremental learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 30534–30544, 2025.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8340–8349, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 787–798, 2014.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv*, 2021.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23638–23647, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022a.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*, 2022b.

Yuheng Lu, Bingshuo Qian, Caixia Yuan, Huixing Jiang, and Xiaojie Wang. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19165–19181, 2025.

Chuancheng Lv, Lei Li, Shitou Zhang, Gang Chen, Fanchao Qi, Ningyu Zhang, and Hai-Tao Zheng. Hyper-lora: Efficient cross-task generalization via constrained low-rank adapters generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16376–16393, 2024.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 565–576, 2021.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 11–20, 2016.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pp. 947–952. IEEE, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.

Yu-Yang Qian, Yuan-Ze Xu, Zhen-Yu Zhang, Peng Zhao, and Zhi-Hua Zhou. Treelora: Efficient continual learning via layer-wise loras guided by a hierarchical gradient-similarity tree. *arXiv preprint arXiv:2506.10355*, 2025.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pp. 146–162. Springer, 2022.

Yihua Shao, Minxi Yan, Yang Liu, Siyu Chen, Wenjie Chen, Xinwei Long, Ziyang Yan, Lei Li, Chenyu Zhang, Nicu Sebe, Hao Tang, Yan Wang, Hao Zhao, Mengzhu Wang, and Jingcai Guo. In-context meta lora generation. In James Kwok (ed.), *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 6138–6146. International Joint Conferences on Artificial Intelligence Organization, 8 2025a. doi: 10.24963/ijcai.2025/683. URL `https://doi.org/10.24963/ijcai.2025/683`. Main Track.

Yihua Shao, Minxi Yan, Yang Liu, Siyu Chen, Wenjie Chen, Xinwei Long, Ziyang Yan, Lei Li, Chenyu Zhang, Nicu Sebe, et al. In-context meta lora generation. *arXiv preprint arXiv:2501.17635*, 2025b.

Yihua Shao, Xiaofeng Lin, Xinwei Long, Siyu Chen, Minxi Yan, Yang Liu, Ziyang Yan, Ao Ma, Hao Tang, and Jingcai Guo. Icm-fusion: In-context meta-optimized lora fusion for multi-task adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8860–8868, 2026.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pp. 1453–1460. IEEE, 2011.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv*, 2023.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23 (7):1661–1674, 2011.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. Screen2words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 498–510, 2021.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, 2023.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119(1):3–22, 2016.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022.

Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. Robustmerge: Parameter-efficient model merging for mllms with direction robustness. *arXiv preprint arXiv:2502.17159*, 2025.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pp. 3987–3995. Pmlr, 2017.